IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application		) PATENT APPLICATION	
Inventors:	David L. Multer, et al.	)	Art Unit: 2175
Application No.:	09/641,028	)	Examiner: Mofiz, Apu M.
Filed:	August 17, 2000	)	
Title:	BASE ROLLING ENGINE FOR DATA TRANSFER AND SYNCHRONIZATION SYSTEM	)	**Customer No. 28554**

DECLARATION OF LEIGHTON RIDGARD
PURSUANT TO 37 C.F.R. §1.131

I, LEIGHTON RIDGARD, declare that:

1. I am an inventor of the invention described and claimed in the above-identified patent application. I am currently an Principal Software Engineer of fusionOne, Inc., assignee of the instant application. I have reviewed the pending application as stated in my Declaration for Patent Application and the pending claims as set forth in the RESPONSE A TO OFFICE ACTION ("RESPONSE A") accompanying this DECLARATION. I have also reviewed U.S. Patent No. 6,295,541 having a filing date of June 7, 1999, and claiming priority to United States Provisional Application Ser. No. 60/069,731, filed Dec. 16, 1997, Ser. No. 60/094,972, filed Jul. 31, 1998, and Ser. No. 60/094,824, filed Jul. 31, 1998.

2. I understand that this Declaration will be filed in the United States Patent and Trademark Office in order to provide factual evidence showing that the invention claimed in the present application was completed prior to the date of December 16, 1997.

3. The facts set forth hereinafter to establish that the claimed invention was completed prior to December 16, 1997 all relate to acts which occurred and were carried out within the United States.

4. I have been personally associated with Rick Onyon since November of 1991. I met Rick in November 1991 when he was looking for an engineer to work on an idea he had for hierarchical storage for PCs. I worked on that product as one of two engineers while I was employed at KnowledgeWare Inc. In February 1992 I left KnowledgeWare to work for Rick's new company Chili Pepper Software. Chili Pepper software was purchased by Cheyenne Software which was later purchased by Computer Associates. I worked with Rick until he left Computer Associates to go to Mango Software in early 1997. Although at this point in time we were working for separate companies, we remained in touch and often spoke about his ideas for starting a new company and a new product to do file synchronization. In February of 1998, I was hired by Mango Software and once again was working with Rick. Rick eventually quit Mango Software, moved to California and started fusionOne. Once funding was in place I quit Mango Software and went to work for Rick in November of 1998 at fusionOne.

5. During all of that time I was living in Atlanta. Rick had moved to New York when Cheyenne acquired Chili Pepper. He would later move to Boston when he went to work for Mango Software. We remained in touch often. Whenever business or personal travel brought him to or through Atlanta

we would get together socially. These meetings almost always involved discussing Rick's latest product ideas.

6. It was at one of these meetings well prior to December of 1997 that Rick introduced me to his latest idea for a file synchronization service. His initial idea was to create a virtual drive on each machine called the "I" drive. The basic idea was that anything you place in that virtual drive would be kept in sync with all other machines also using the "I" drive. Rick would later decide to call this product "OneDrive", had decided on a company name and had registered domain names for this name. None of these product or domain names were used because by then the product had evolved into so much more.

7. Admittedly I was not initially impressed with the OneDrive idea. Nevertheless, we continued to have conversations about how to sell it, how to use FTP storage, partnering with ISPs, and other aspects of the system.

8. Within a few weeks of our initial conversation on this idea, and still well prior to December 1997, Rick realized that he wanted to synchronize more than just his files. He also wanted his emails, calendar items, and contacts to be kept in sync between his home and work PCs. I offered that we could simply keep the outlook PST in sync with the same techniques we were planning to use to keep the I-Drive files in sync. The outlook PST tends to be much larger than the typical files we had expected to keep in sync. This quickly led us to the conclusion that our sync engine needed to be efficient about how much data it transmitted between PCs to keep them in sync. I speculated that we could use some sort of binary differencing methodology to generate small delta files between a

current PST and a historical PST. Such technology was available and we would either incorporate an existing one or write our own.

9. It became apparent to us that the binary differencing idea did not make any sense when applied to a PST. If we used this method then we would be producing an exact duplicate of the Outlook PST on the second machine. This was a problem because: (1) a user may not want their personal family information stored on their home PC to be synced to their work PC; (2) sensitive office data in emails should not be synced to the home PC, and (3) we would not be able to support a situation where the two PCs were not running the same version of Outlook. We also realized that a huge beneficial side-effect of creating a system extracting specific information from Outlook was that would be able to extend it to keep two totally different personal information managers or "PIMs" in sync. At this point we decided that binary differencing would only be used for generic files but not for PIM data.

10. We needed to create a technology that would allow us to extract the specific information in one PIM that the user specified, encode it efficiently, transfer it to another machine, and then apply it to the destination PIM. Outlook was our primary focus, but we were mindful of other PIMs while we pursued this goal. Rick established the guideline that the solution needs to be able to sync data in Outlook on one PC into an empty copy of Outlook on another PC with no loss of data. Furthermore successive syncs should be accomplished with the minimal amount of data transfer.

11. I thought about this a lot and eventually with Rick came up with the model of recording transactions on the source machine which would be played back on the target machine. In keeping

with the original file sync idea we kept in place the idea of using a temporary holding place (such as FTP storage at one's ISP) for these transaction logs. This would eventually be called store-and-forward sync.

12. From this point forward, and still prior to December 1997, we had a clear direction of every next step to be followed.

13. Next, and still prior to December 1997, we went about devising a quick way to generate the transactions. Simply looking at a PST would not tell us what was different since the last time we synced. We quickly hit upon two ideas. One was to monitor the PIM and record the user activity in the transaction log. The other was to store a historical copy of the PIM data and then compare the current PIM data with the historical copy and generate transactions from the difference. It was quickly decided that we would do both. We would use monitoring for PIMs that could support it and we would use historical comparison for all other PIMs.

14. Mostly because of Rick's insisting that we support every field in Outlook and all other PIMs (current and future), I was driven to device a database scheme that could be used to store the historical data. This would eventually be known in our product as the AOS (Application Object Store). I concluded that for example: a contact from Outlook with a certain set of fields and a contact from Organizer which may contain a different set of fields needed to be stored in the same database. This led to the design of creating a database that was a storage facility for 'fields' instead of the usual database methodology of storing 'records'. The result was a database that could contain the superset of all PIM data across all applications current and future.

15. During the following months all prior to December 1997, Rick and I would continue to develop these ideas over the phone and on his occasional visits to Atlanta. This included devising a procedure for generating the transaction logs from the difference between the current PIM data and the data stored in the AOS. We also wanted to ensure that files could still be kept in sync using the same or very similar technique. We came up with the idea that we needed a differencing engine that could compute the differences of not only PIM data but binary files as well. This differencing engine would eventually be known as Structured Delta. The key design feature we came up with at the time was that the differencing engine would not actually perform a sync per se. Instead the differencing engine had a dual role. One was to determine what changed, generate transaction logs (later to be called change logs), and transmit them to the central store (FTP site). The other role was the reverse. It would get the logs from the central store, decode the logs, and apply the transactions to the target PIM. Sync was actually accomplished by the sequential use of the differencing engine on the source and then target machines.

16. The final task was to design the transaction logs themselves. This was fairly straight forward given our database design of storing items at the field level. The differencing engine would determine what was added, moved, renamed, changed, or deleted and would do this at the field level. It would generate into our change log the list of operations (add, delete, move, etc.) along with the list of fields that were associated with that operation. A header would be placed at the beginning of the file to include items like the schema version, file signature, date generated, PIM used, device used etc. This was done based on our experience as programmers. Schema version would be used so we could revise the format in the future if required. The signature was used so the file integrity could

be validated before use etc. These things are common in most file formats and we decided to do likewise in our new file format.

17. Still well prior to December 1997, we had a concrete idea of the aforementioned aspects of the system. Further discussions would lead us to a couple more important enhancements to our design.

18. The first enhancement was that we could treat file synchronization just like PIM sync. Files were just like all other data in our model because they could be thought of as consisting of a set of fields (name, date, size, body, etc.). However we decided to keep the original idea of using binary differencing for the file body. So we incorporated into Structured Delta the methodology of encoding into Change logs the data generated from the binary difference between the current file and the historical copy.

19. The second enhancement came from mental field tests of the now rapidly evolving sync product design. We played out the scenario of a busy user who syncs a lot. We realized that such a user would have stored in their FTP site many Change logs that contained historical data. For instance if the contact had been deleted the old change logs would still contain historical useless information about that contact. In order to use the user space more responsibly we decided that periodically we could run a process against the central store that would collapse all of the change logs into one. Initially we thought about running this process on the server without even involving the user. Such a process would apply successive change logs to each other and combine changes and deletes etc. to produce a minimal set of resultant changes collapsed into ideally one change log.

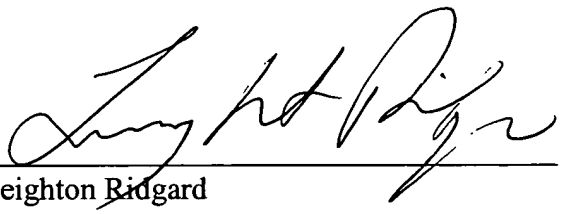
During these discussions we decided that even if not done behind the scenes on the server this feature would be equally valuable if implemented on the client. We later decided to call server-side log rolling "Base rolling" and left the term "log rolling" for use in describing client-side rolling. The term "base" was used to denote the concept that this would establish a base starting point for future changes. In other words, that Base log combined with future change logs would comprise all of one's PIM data. When base rolling was done in the future it would then establish a new base.

20. Due to my responsibilities at my place of employment, I did not have significant time to work on the next step which was to build a prototype. I recall that at Rick's request I began building a prototype in February 1998. I understand that Rick believes this occurred earlier. In the interest of time, it was decided to produce a proof of concept prototype which synced the entire PST from one device to the other. Although this was certainly not the implementation which we had decided upon, it was sufficient as a proof of concept. In the spring of 1998, Liam Stannard and Rob Garner started working with me to produce the second prototype. The second prototype was completed in the summer of 1998 and implemented an early version of our change log and structured delta design.

21. In the summer of 1998, David Multer, Rob Garner, Liam Stannard, Don Cash, and myself met in Atlanta to produce the official architecture of the product. This occurred rapidly due to the concrete conception we had brought to that meeting. In November of 1998 we started working for fusionOne implementing that design in a commercial product.

23. Throughout our development of our system, we worked diligently to reduce the invention to practice. This included the creation of at least three (3) prototypes throughout the process of hiring engineers and gaining funding to complete development of the system.

24. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: JULY 16TH 2007 By: 

Leighton Ridgard